



From Reeds and Shepp's to continuous-curvature paths

Thierry Fraichard, Alexis Scheuer

► To cite this version:

Thierry Fraichard, Alexis Scheuer. From Reeds and Shepp's to continuous-curvature paths. IEEE Transactions on Robotics, 2004, 20 (6), pp.1025-1035. 10.1109/TRO.2004.833789 . inria-00000009

HAL Id: inria-00000009

<https://inria.hal.science/inria-00000009>

Submitted on 10 May 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From Reeds and Shepp's to Continuous-Curvature Paths

Thierry Fraichard^a and Alexis Scheuer^b

Inria^c Rhône-Alpes & Gravir^d-CNRS^e

Final version - September 22, 2004

Abstract — This paper presents CC Steer, a steering method for car-like vehicles, *ie* an algorithm planning paths in the absence of obstacles. CC Steer is the first to compute paths with (a) continuous curvature, (b) upper-bounded curvature, and (c) upper-bounded curvature derivative. CC Steer also verifies a topological property that ensures that when it is used within a general motion planning scheme, it yields a complete collision-free path planner. The coupling of CC Steer with a general planning scheme yields a path planner that computes collision-free paths verifying the properties mentioned above. Accordingly, a car-like vehicle can follow such paths without ever having to stop in order to reorient its front wheels. Besides, such paths can be followed with a nominal speed which is proportional to the curvature derivative limit. The path computed by CC Steer are made up of line segments, circular arcs and clothoid arcs. They are not optimal in length. However, it is shown that they converge toward the optimal 'Reeds and Shepp' paths when the curvature derivative upper-bound tends to infinity. The capabilities of CC Steer to serve as an efficient steering method within two general planning schemes are also demonstrated.

Keywords — Nonholonomic vehicles, path planning, continuous-curvature paths.

Acknowledgements — This work was supported by the French Praxitéle programme <<http://www-rocq.inria.fr/praxitele>>, the European Inco-Copernicus project "Multi-agent robot systems for industrial applications in the transport domain" <<http://wwwipr.ira.uka.de/kamara/projects/index1.html>>, and the French programme "La Route Automatisée": <<http://www.lara.prd.fr>>.

^aThierry Fraichard is a Research Associate at Inria.

^bAlexis Scheuer is an Associate Professor at the University Henry Poincaré, Nancy (FR). E-mail: alexis.scheuer@loria.fr

^cNational Institute for Research in Computer Science and Control

^dGraphics, Vision and Robotics Lab.

^eNational Centre for Scientific Research.

From Reeds and Shepp's to Continuous-Curvature Paths

Thierry Fraichard and Alexis Scheuer

Abstract—This paper presents CC Steer, a steering method for car-like vehicles, *ie* an algorithm planning paths in the absence of obstacles. CC Steer is the first to compute paths with (a) continuous curvature, (b) upper-bounded curvature, and (c) upper-bounded curvature derivative. CC Steer also verifies a topological property that ensures that when it is used within a general motion planning scheme, it yields a complete collision-free path planner. The coupling of CC Steer with a general planning scheme yields a path planner that computes collision-free paths verifying the properties mentioned above. Accordingly, a car-like vehicle can follow such paths without ever having to stop in order to reorient its front wheels. Besides, such paths can be followed with a nominal speed which is proportional to the curvature derivative limit. The path computed by CC Steer are made up of line segments, circular arcs and clothoid arcs. They are not optimal in length. However, it is shown that they converge toward the optimal ‘Reeds and Shepp’ paths when the curvature derivative upper-bound tends to infinity. The capabilities of CC Steer to serve as an efficient steering method within two general planning schemes are also demonstrated.

Index Terms—Nonholonomic vehicles, path planning, continuous-curvature paths.

I. INTRODUCTION

Ever since Laumond’s pioneering paper in 1986 [1], a lot of research works have addressed collision-free path planning for nonholonomic systems in general and car-like vehicles in particular. Nonholonomic systems are subject to kinematic constraints that restrict their admissible directions of motion. Nonholonomy makes path planning more difficult since the paths planned must take into account the constraints imposed both by the obstacles and the nonholonomic constraints (the reader is referred to [2] for a recent and extensive review on this topic). Car-like vehicles are archetypal nonholonomic systems: they can only move forward or backward in a direction perpendicular to the orientation of their rear wheels axle; besides their turning radius is lower bounded because of the mechanical limits on the steering angle.

The review of the research works that plan collision-free paths for car-like vehicles shows that most of them use a “simplified” model for the car-like vehicle (one with only three configuration parameters in which the control is the angular velocity), and compute planar paths made up of line segments connected with tangential circular arcs of minimum radius, *eg* [3], [4], [5], [6], etc. The reason for this must be that the shortest path between two configurations for the simplified car

is such a path (result established first by Dubins [7] for the car moving forward only, and later by Reeds and Shepp [8] for the car moving both forward and backward).

However, the curvature of this type of path is discontinuous: discontinuities occur at the transitions between segments and arcs and between arcs with opposite direction of rotation. The curvature being related to the front wheels’ orientation, if a real car were to track precisely such a type of path, it would have to stop at each curvature discontinuity so as to reorient its front wheels. Curvature continuity is therefore a desirable property. Besides, since the derivative of the curvature is related to the steering velocity of the car, it is also desirable that the derivative of the curvature be upper-bounded so as to ensure that such paths can be followed with a given speed (proportional to the curvature derivative limit). Ref. [9] was the first to recognise this issue and to address the problem of computing continuous-curvature paths with upper-bounded curvature derivative; it introduced a model for the car-like vehicle moving forward only with curvature as an extra configuration parameter in which the control is the angular acceleration rather than the angular velocity. The work presented herein follows in [9]’s footsteps: it also addresses the problem of planning paths with (a) continuous-curvature and (b) upper-bounded curvature derivative for car-like vehicles. In addition to that however, it considers (c) upper-bounded curvature, (d) forward and backward motions, and (e) collision avoidance.

The approach we have chosen to solve the problem at hand relies upon the design of a steering method, *ie* an algorithm that computes a path between two configurations in the absence of obstacles. Given such a steering method, it is possible to use it within a general motion planning scheme such as the Probabilistic Path Planner [10], the Ariadne’s Clew Algorithm [11] or the Holonomic Path Approximation Algorithm [12], in order to deal with the obstacles and solve the full problem (in these schemes, the steering method is used along with a collision checker to connect pairs of selected configurations).

The steering method is a key component in these planning schemes and the main contribution of this paper is the first steering method that computes paths with (a) *continuous-curvature*, (b) *upper-bounded curvature* and (c) *upper-bounded curvature derivative* for car-like vehicles. Our steering method is *topologically admissible*, *ie* it verifies a topological property [13] that ensures that the coupling with one of the aforementioned planning scheme yields a complete (or probabilistically complete) collision-free path planner. Unlike [9] or [14], the focus of our work is not on optimal path planning

Thierry Fraichard is a Research Associate at the National Institute for Research in Computer Science and Control (Inria), Rhône-Alpes Research Centre, Grenoble (FR). E-mail: thierry.fraichard@inria.fr.

Alexis Scheuer is an Associate Professor at the University Henry Poincaré, Nancy (FR). E-mail: alexis.scheuer@loria.fr

and our steering method does not compute minimal length paths. However, it computes paths whose length is close to the length of the optimal paths for the simplified car (as a matter of fact, when the curvature derivative limit tends to infinity, the paths computed become Reeds and Shepp paths).

There is a number of research works that are relevant to the design of a steering method such as ours; they are reviewed in §II. Then the model of the car is detailed in §III while the steering method proposed is described in §IV. Finally §V presents experimental results for the steering method on its own and embedded in two general planning schemes.

II. RELATED WORKS

A first class of works related to the problem of computing continuous-curvature paths falls into the “path smoothing” class: the focus is on turning a sequence of configurations (or a nominal route such as a polygonal line) into a smooth curve that is then passed to the control system of the vehicle. The curves used fall into two categories: (1) curves whose coordinates have a closed-form expression, *eg* B-splines [15], quintic polynomials [16] or polar splines [17], and (2) parametric curves whose curvature is a function of their arc length, *eg* clothoids [18], cubic spirals [19], quintic G^2 -splines [20], or intrinsic splines [21]. These approaches are interesting. Unfortunately issues such as completeness or topological admissibility are completely ignored. They usually make simplifying assumptions (on the respective positions of the configurations that are to be connected, on the type of path generated, etc.) that render them unsuitable for our purpose.

More relevant to our problem are the works stemming from the recent application of control theory tools and ideas to study the controllability of nonholonomic systems in order to derive novel steering methods.

First, optimal control theory could be used to solve the problem at hand [22]. Indeed optimal length paths would be the paths of choice for a steering method. Unfortunately, optimal paths are extremely difficult to characterise in general. This characterisation is available for the simplified car-like system [8], but not for the system considered herein. Besides there are indications that the optimal paths for the system considered herein are in general irregular and cannot be used in practice since they involve infinite chattering (*cf* §III-B). Nevertheless, it would be possible, in theory, to compute an approximation of the optimal paths using a numerical optimisation method such as the one presented in [23]. In practice however, such a method poses a number of problems (*cf* the experimental results reported in [24]). Among them, the fact that the convergence to the optimum is not guaranteed meaning that the topological admissibility is not guaranteed either.

Second, given that the car-like vehicle is similar (from a control point of view) to the system made up of a differential drive vehicle pulling one trailer, it could be possible to use one of the steering methods that were proposed for such a system. These steering methods exploit different properties of such a system, namely nilpotence [25], chained form [26] and differential flatness [27]. We briefly review them now. The reader

is referred to [28] for underlying theoretical details (about these properties in particular), and to [24] for an experimental comparison between these steering methods when applied to the case of a differential drive vehicle pulling several trailers.

a) Nilpotence: nilpotent systems were first studied in [25]. For such systems, it is possible to compute piecewise constant controls steering the system exactly to the goal. The car-like vehicle is not nilpotent but it is possible to compute a nilpotent approximation of such a system [29] and use it to steer the vehicle. Unfortunately, the goal configuration would never be reached exactly.

b) Chained form: as for systems that can be converted into chained form [26], it is possible to steer them exactly to the goal using either sinusoidal [26], polynomial [30] or piecewise constant [31] controls. For these steering methods, path planning takes place in a space defined by parameters whose physical meaning is unclear. The relationship between the shape of the path in the parameter space and the shape of the corresponding path in the actual workspace is a very complex one to the extent that it is extremely difficult to obtain topological admissibility (*cf* [24]).

c) Differential flatness: it is a property enjoyed by certain mechanical systems which was first studied in [27]. In this case too, path planning takes place in a space defined by parameters different, in general, from the configuration parameters. Exploiting this property, [32] designed an efficient steering method for differential drive vehicles pulling trailers, method that was later adapted to the case of a car-like vehicle [33]. This steering method, developed in parallel with the one presented in this paper, is very close to being a good candidate to solve our problem: it is topologically admissible and it computes continuous-curvature paths for car-like vehicles. However, it does not take into account the upper-bounded curvature derivative constraint nor the upper-bounded curvature constraint (what it does is to compute a path first and then check afterwards that the upper-bounded curvature constraint is not violated).

III. STATEMENT OF THE PROBLEM

A. Model of the Car-Like Vehicle

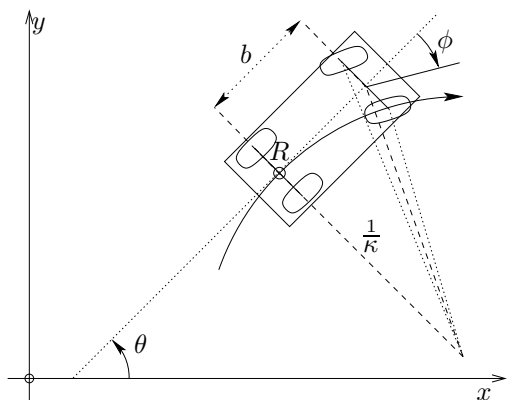


Fig. 1. The car-like vehicle \mathcal{A} : $R = (x, y)$ is the reference point and θ the main orientation. ϕ is the steering angle and b the wheelbase.

As mentioned earlier, in order to address the curvature discontinuity problem, [9] introduced the following model for a car-like vehicle \mathcal{A} moving forward only (Fig. 1):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \sigma \quad (1)$$

This model extends the well-known Dubins model [7] by considering the car as a four-dimensional system with the curvature κ as an extra configuration parameter (the three other configuration parameters are (x, y) the coordinates of the reference point R and θ the main orientation of \mathcal{A}). This system has two controls: (1) v , the driving velocity of the rear wheels, and (2) σ , the angular acceleration which is related to $\dot{\phi}$, the steering velocity of the front wheels. Let b denote the wheelbase of \mathcal{A} , the following relationships hold:

$$\kappa = \frac{\tan \phi}{b}, \quad \sigma = \dot{\kappa} = \frac{\dot{\phi}}{b \cos^2 \phi} \quad (2)$$

The focus in [9] is on shortest path planning. A constant unit driving velocity is assumed along with an upper-bound on the angular acceleration (stemming from the fact that the steering velocity of an actual car is limited), in other words:

$$v = 1 \text{ and } |\sigma| \leq \sigma_{\max} \quad (3)$$

Under these assumptions, planning a trajectory between two configurations is equivalent to computing a continuous-curvature planar curve, with a bound on the curvature derivative, between two points in the plane with prescribed tangents and curvatures.

This model bounds the angular acceleration σ rather than the steering velocity $\dot{\phi}$ (although it is the steering velocity of \mathcal{A} which is physically limited). This choice is in fact conservative, it ensures that the steering velocity limit is never violated no matter what the value of ϕ is (indeed $\forall \phi : \dot{\phi} \leq \dot{\phi} / \cos^2 \phi$).

For our purpose, we keep [9]’s model and the constant unit driving velocity assumption. However, for further realism, we enrich the model as follows. First, to allow backward motions, v can now take two values: $v = 1$ (forward motion) and $v = -1$ (backward motion). In other words, cusp points are now allowed. At a cusp, \mathcal{A} instantaneously changes its motion direction, both its orientation and curvature remain continuous. Second, given that κ is related to the steering angle ϕ and that ϕ is mechanically limited, $|\phi| \leq \phi_{\max}$, a bounded curvature constraint is introduced:

$$|\kappa| \leq \kappa_{\max} = \tan \phi_{\max} / b \quad (4)$$

Accordingly, our final model for \mathcal{A} is given by (1) with:

$$|\kappa| \leq \kappa_{\max}, |v| = 1 \text{ and } |\sigma| \leq \sigma_{\max} \quad (5)$$

Henceforth, the term *CC Car* (for continuous-curvature car) is used to denote a vehicle with such a model. Planning a trajectory for the CC Car between two configurations is equivalent to computing a continuous-curvature planar curve joining two points in the plane with prescribed tangents and

curvatures. Such a curve must verify the upper-bounds on its curvature and curvature derivative.

By design, such a curve can be followed at unit speed by \mathcal{A} . To enable \mathcal{A} to follow it with a different constant speed v_d , the curvature derivative limit should be set to $\sigma_{\max} v_d^{-1}$.

B. Properties of the CC Car

Following the study of model (1) carried out by [9], [14] and [34], the CC Car model (1)+(5) was studied by Scheuer in [35] and [36]. It is established that the CC Car is small-time controllable [35, Theorem 1]. The set of configurations reachable from any configuration q before a time t contains a neighbourhood of q for any t . The condition of existence of the optimal, *ie* shortest, paths is also established: in the absence of obstacles, if a path exists between two configurations then an optimal path exists [35, Theorem 2].

The nature of the optimal paths is more difficult to establish. However, [36] demonstrates that, for the CC Car moving forward only ($v = 1$), the optimal paths are made up of: (a) line segments, (b) circular arcs of radius κ_{\max}^{-1} , and (c) clothoid arcs¹ of sharpness $\pm \sigma_{\max}$. It also demonstrates that, whenever the shortest path includes a line segment (which is the case as soon as the distance between the start and end configurations is large enough), it involves infinite chattering: in other words, it contains an infinite number of clothoid arcs that accumulate towards the endpoints of the segment (these results are the extension of similar results obtained in the case of the model (1) by [9] and [14]).

Characterising the true nature of the optimal paths for the CC Car is beyond the scope of this paper. Based on the results mentioned above, it is conjectured that they will (at least) be made up of line segments, circular arcs and clothoid arcs, and that they will be irregular. This prevented us from designing a steering method computing the optimal path between two configurations. Instead, it prompted us to settle for a steering method computing paths essentially made up of locally optimal paths, *ie* line segments, circular arcs of radius κ_{\max}^{-1} , and clothoid arcs of sharpness $\pm \sigma_{\max}$.

IV. STEERING THE CC CAR

A. Principle

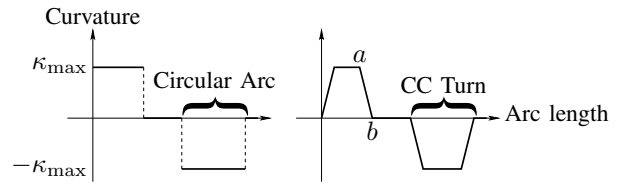


Fig. 2. Discontinuous curvature profile of a RS Path (left) vs piecewise continuous curvature profile of a CC Path (right). The part from a to b is a clothoid arc.

The steering method we have designed is called *CC Steer*. It computes paths called *CC Paths* that are derived from “Reeds and Shepp’s paths” (RS Paths) [8].

¹A clothoid is a curve whose curvature varies linearly with its arc length: $\kappa(s) = \alpha s + \kappa(0)$, α is the *sharpness* of the clothoid.

The RS Path between two configurations is the shortest path made up of line segments and circular arcs of minimum radius κ_{\max}^{-1} . Its curvature profile looks like the one depicted in Fig. 2. CC Paths are similar to RS Paths but, in order to ensure curvature continuity, the circular arcs are replaced by special transitions called *CC Turns* whose purpose is to change the CC Car's orientation. Their curvature varies continuously and piecewise linearly from 0 up and then down back to 0. They are made up of circular arcs of minimum radius κ_{\max}^{-1} and clothoid arcs of sharpness $|\sigma| \leq \sigma_{\max}$ (Fig. 2).

CC Turns and line segments are combined in order to form the CC Paths (cf §IV-B and §IV-C). However, the CC Paths obtained by combining CC Turns and line segments only cannot account for the small-time controllability of the CC car. To avoid this problem, an extra type of CC Paths made up only of line segments and clothoid arcs of sharpness $|\sigma| \leq \sigma_{\max}$ is introduced (cf §IV-D).

B. CC Turns

1) *General Case*: In general, a CC Turn is made up of three parts: (a) a clothoid arc of sharpness $\sigma = \pm\sigma_{\max}$ whose curvature varies from 0 to $\pm\kappa_{\max}$, (b) a circular arc of radius $\pm\kappa_{\max}^{-1}$, and (c) a clothoid arc of sharpness $-\sigma$ whose curvature varies from $\pm\kappa_{\max}$ to 0.

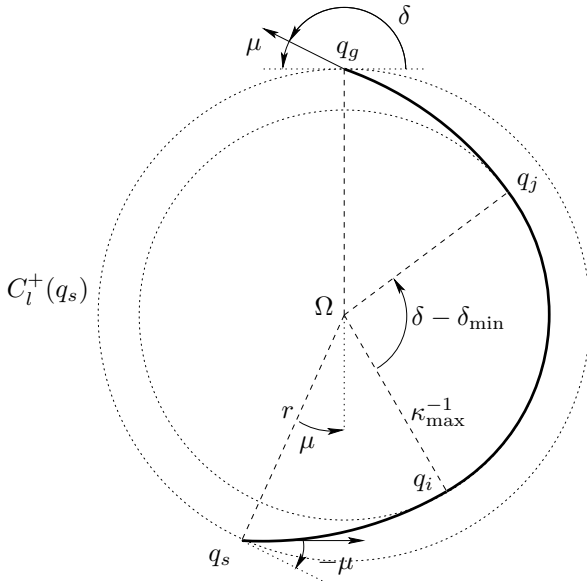


Fig. 3. CC Turns: general case.

What happens when the CC Car follows a CC Turn is illustrated in Fig. 3. Let $q_s = (x_s, y_s, \theta_s, 0)$ be the start configuration. Without loss of generality, it is assumed that $q_s = (0, 0, 0, 0)$ and that the CC Car moves forward while turning to the left. First, it follows a clothoid arc of length $\kappa_{\max}/\sigma_{\max}$ and sharpness σ_{\max} until it reaches q_i :

$$q_i = \begin{cases} x_i &= \sqrt{\pi/\sigma_{\max}} C_f(\sqrt{\kappa_{\max}^2/\pi\sigma_{\max}}) \\ y_i &= \sqrt{\pi/\sigma_{\max}} S_f(\sqrt{\kappa_{\max}^2/\pi\sigma_{\max}}) \\ \theta_i &= \kappa_{\max}^2/2\sigma_{\max} \\ \kappa_i &= \kappa_{\max} \end{cases} \quad (6)$$

with C_f and S_f , the Fresnel integrals. Then it follows a circular arc of radius κ_{\max}^{-1} until it reaches $q_j = (x_j, y_j, \theta_j, \kappa_{\max})$. The centre of this circular arc, Ω , is located at distance κ_{\max}^{-1} from q_i in the direction normal to θ_i :

$$\Omega = \begin{cases} x_{\Omega} = x_i - \kappa_{\max}^{-1} \sin \theta_i \\ y_{\Omega} = y_i + \kappa_{\max}^{-1} \cos \theta_i \end{cases} \quad (7)$$

Finally it follows a clothoid arc of sharpness $-\sigma_{\max}$ until it reaches the goal configuration $q_g = (x_g, y_g, \theta_g, 0)$. Let $\delta = (\theta_g - \theta_s) \bmod 2\pi$ denote the change of orientation between q_s and q_g . δ is the *deflection* of the CC Turn, it is used to characterise CC Turns. The deflection of the CC Turn whose circular arc has zero length is $\delta_{\min} = \kappa_{\max}^2 \sigma_{\max}^{-1}$.

It is the angular value of the circular arc of a CC Turn, ie $\delta - \delta_{\min}$, that actually determines where the goal configuration is: the locus of the goal configurations is a circle $C_l^+(q_s)$, henceforth called a *CC Circle*, whose centre is Ω and whose radius r is:

$$r = \sqrt{x_{\Omega}^2 + y_{\Omega}^2} \quad (8)$$

In addition, the angle μ between the orientation of q_g and the tangent to $C_l^+(q_s)$ at q_g is constant; it is the opposite of the angle between the orientation of q_s and the tangent to $C_l^+(q_s)$ at q_s :

$$\mu = \text{atan}(x_{\Omega}/y_{\Omega}) \quad (9)$$

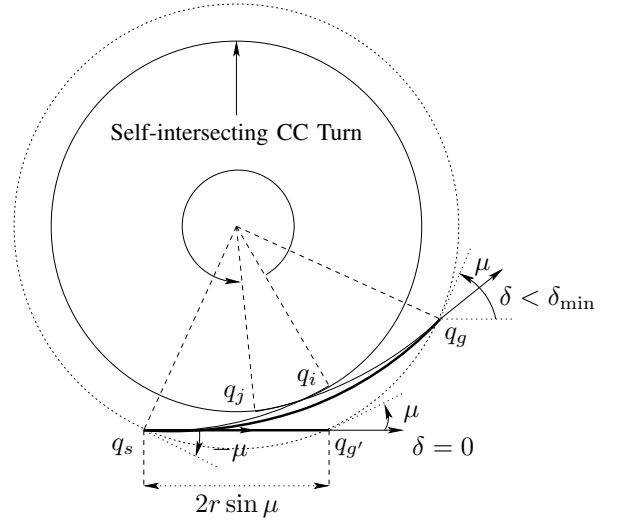


Fig. 4. CC Turns: “ $\delta = 0$ ” and “ $0 < \delta < \delta_{\min}$ ” cases.

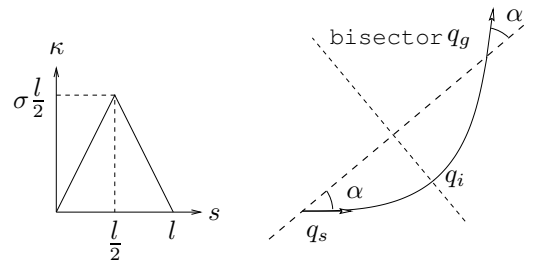


Fig. 5. Curvature profile and track of an elementary path of sharpness σ and length l [37].

2) *CC Turns of Small and Large Deflections*: with the definition above, a CC Turn makes a loop and intersects itself when the angle of its circular arc becomes too important (Fig. 4). It can be shown that self-intersection happens for deflections $0 < \delta < \delta_{\min}$. In this case, we propose to use instead a loopless and shorter path called an *elementary path* and made up of: (a) a clothoid arc of sharpness $\sigma \leq \sigma_{\max}$ and (b) a symmetric clothoid arc of sharpness $-\sigma$ (Fig. 5). Such a path is feasible and we have shown in [38] that there is a unique $\sigma \leq \sigma_{\max}$ such that q_g belongs to $C_l^+(q_s)$:

$$\sigma = \frac{\pi(\cos(\delta/2)C_f(\sqrt{\delta/\pi}) + \sin(\delta/2)S_f(\sqrt{\delta/\pi}))^2}{r^2 \sin^2(\delta/2 + \mu)} \quad (10)$$

As for the $\delta = 0$ case, the CC Turn reduces to the line segment of length $2r \sin \mu$ so as to ensure that the goal configuration also belongs to $C_l^+(q_s)$ (Fig. 4).

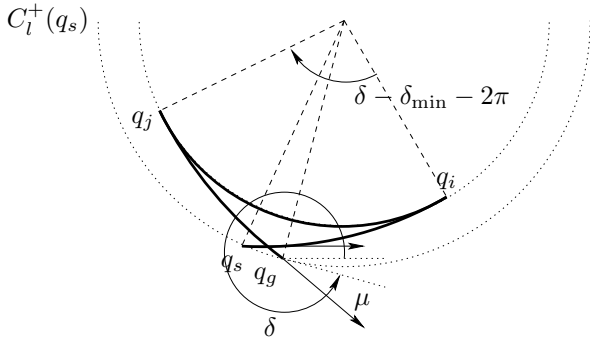


Fig. 6. CC Turns: “ $\delta_{\min} + \pi \leq \delta < 2\pi$ ” case.

Finally, since the CC Car can make backward and forward motions, it is possible to further refine CC Turns so as to reduce their length. Once q_i is reached, it is shorter for CC Turns of large deflection to back up to q_j instead of moving forward (Fig. 6). Given that the angle of the circular arc of a CC Turn of deflection δ is $\delta - \delta_{\min}$ when the CC Car moves forward from q_i to q_j , and $\delta - \delta_{\min} - 2\pi$ when it moves backward, the backward motion is shorter when $\delta \geq \delta_{\min} + \pi$.

3) *Properties of the CC Turns*: In summary, a CC Turn allows the CC Car to reach any goal configuration q_g with a null curvature which is located on the circle $C_l^+(q_s)$ and such that the angle between the orientation of q_g and the tangent to $C_l^+(q_s)$ at q_g is constant. It is the deflection associated with q_g that determines the nature of the CC Turn:

- $\delta = 0$: the CC Turn is a line segment.
- $0 < \delta < \delta_{\min}$: the CC Turn is made up of (a) a clothoid arc of sharpness $\sigma \leq \sigma_{\max}$ and (b) a symmetric clothoid arc of sharpness $-\sigma$.
- $\delta_{\min} \leq \delta < \delta_{\min} + \pi$: the CC Turn is made up of (a) a clothoid arc of sharpness σ_{\max} whose curvature varies from 0 to κ_{\max} , (b) a *forward* circular arc of radius κ_{\max}^{-1} , and (c) a clothoid arc of sharpness $-\sigma_{\max}$ whose curvature varies from κ_{\max} to 0.
- $\delta_{\min} + \pi \leq \delta < 2\pi$: the CC Turn is made up of (a) a clothoid arc of sharpness σ_{\max} whose curvature varies from 0 to κ_{\max} , (b) a *backward* circular arc of radius κ_{\max}^{-1} , and (c) a clothoid arc of sharpness $-\sigma_{\max}$ whose curvature varies from κ_{\max} to 0.

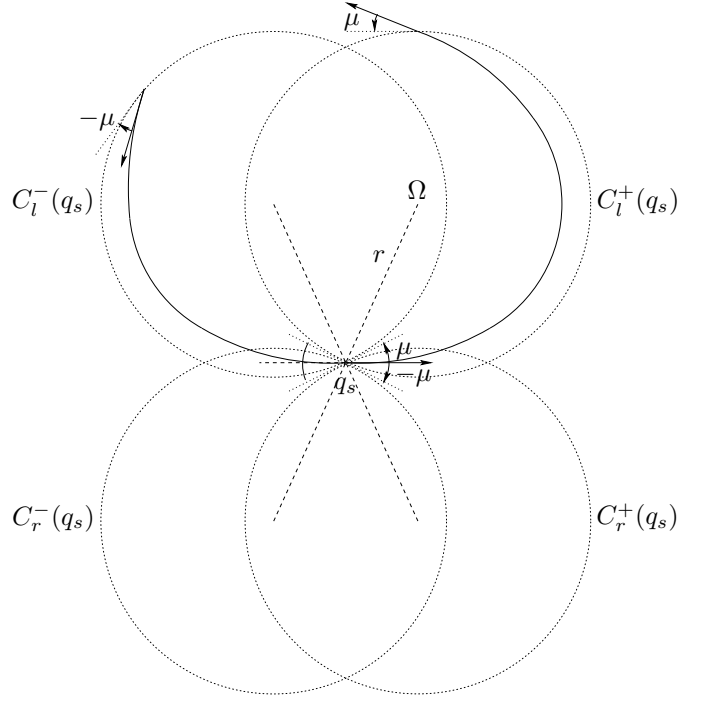


Fig. 7. The four CC Circles $C_l^+(q_s)$, $C_r^+(q_s)$, $C_l^-(q_s)$ and $C_r^-(q_s)$ attached to q_s .

The above analysis was carried out for the case of the CC Car moving forward while turning to the left. The case where the CC Car is turning to the right is dealt with in the same manner, it yields a symmetric CC Circle $C_r^+(q_s)$ (Fig. 7). Two similar CC Circles, $C_l^-(q_s)$ and $C_r^-(q_s)$, are obtained when the CC Car moves backward (Fig. 7).

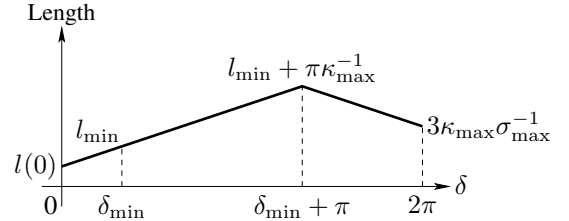


Fig. 8. Arc length of a CC Turn ($\kappa_{\max} = 1, \sigma_{\max} = 1$) as a function of δ .

The arc length of a CC Turn depends upon its nature. Let $l(\delta)$ denote the arc length of a CC Turn of deflection δ , it is defined as follows (Fig. 8):

- $\delta = 0$: the CC Turn is a line segment of length $l(0) = 2r \sin \mu$.
- $0 < \delta < \delta_{\min}$: let σ be the sharpness characterising the CC Turn in this case: $l(\delta) = 2\sqrt{\delta\sigma^{-1}}$. $l(\delta)$ increases monotonously from $2r \sin \mu$ to $l_{\min} = 2\kappa_{\max}\sigma_{\max}^{-1}$. l_{\min} is the arc length of the general CC Turn whose circular arc has zero length.
- $\delta_{\min} \leq \delta < \delta_{\min} + \pi$: in this case, the arc length of the CC Turn is l_{\min} plus the arc length of its circular arc: $l(\delta) = l_{\min} + (\delta - \delta_{\min})\kappa_{\max}^{-1}$. $l(\delta)$ increases linearly from l_{\min} to $l_{\min} + \pi\kappa_{\max}^{-1}$.

- $\delta_{\min} + \pi \leq \delta < 2\pi$: this is the same case as above:
 $l(\delta) = l_{\min} + (2\pi - \delta + \delta_{\min})\kappa_{\max}^{-1}$. $l(\delta)$ decreases linearly
from $l_{\min} + \pi\kappa_{\max}^{-1}$ to $l_{\min} + \delta_{\min}\kappa_{\max}^{-1} = 3\kappa_{\max}\sigma_{\max}^{-1}$.

C. From CC Turns to CC Paths

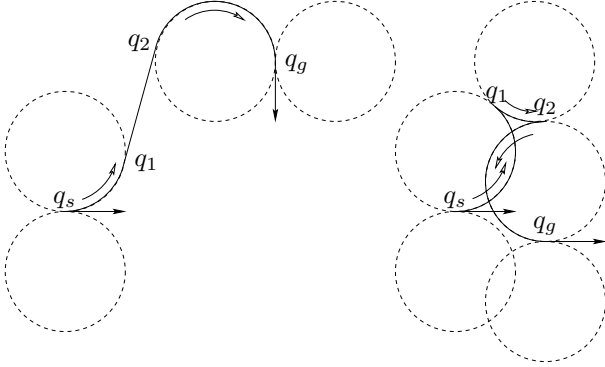


Fig. 9. How RS Paths are computed: two circles of radius κ_{\max}^{-1} are attached to the start and goal configurations q_s and q_g . Then a tangent intermediate line segment (or circular arc of radius κ_{\max}^{-1}) $q_1 q_2$ is used to connect two of these circles.

CC Steer computes CC Paths by combining CC Turns and line segments the same way circular arcs and line segments are combined to form RS Paths (a basic illustration of how RS Paths are computed is depicted in Fig. 9). Accordingly, CC Steer has to analyse the tangency relationships that may exist between the two sets of four CC Circles attached to the start and goal configurations, and line segments or other similar CC Circles. However, due to the fact that the orientation of the configurations located on the CC Circles make a constant angle μ with the tangent to these circles, the tangency relationships considered here (henceforth denoted by μ -tangency) is slightly different from the classical one. Accordingly, before detailing how CC Steer operates (§IV-C.3), we present the particulars of the μ -tangency. Two cases are considered: μ -tangency between line segments and CC Circles (§IV-C.1), and μ -tangency between CC Circles (§IV-C.2).

1) *Line Segments-CC Circles μ -Tangency*: In Reeds and Shepp's case, two circles associated with two given configurations can be connected by one of the tangent line segments existing between them. In our case however, the μ -tangent line segments are different: they must cross the CC Circles so as to make an angle μ with the tangent at the intersection points.

As in the regular tangent case, μ -tangent line segments can be internal (*ie* separating) or external depending on the type of CC Circles that are to be connected. Fig. 10 illustrates how μ -tangent line segments are obtained in two cases (the other cases are symmetric and derived similarly):

- **Connecting C_r^+ and C_l^-** : the μ -tangent line segment is external and parallel to the line of centres $\Omega^1 \Omega^2$ (Fig. 10 top). A straightforward geometric analysis shows that the external μ -tangent line segment $q_1 q_2$ exists iff:

$$l(\Omega^1 \Omega^2) \geq 2r \sin \mu \quad (11)$$

and that its length is:

$$l(q_1 q_2) = l(\Omega^1 \Omega^2) - 2r \sin \mu \quad (12)$$

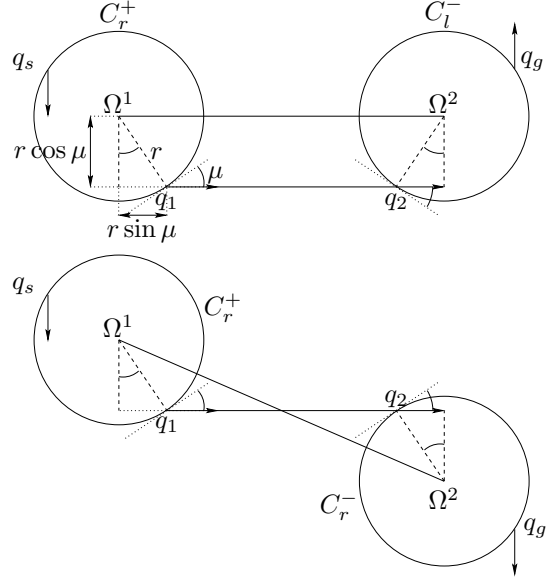


Fig. 10. μ -tangent line segments between two CC Circles, external (top) and internal (bottom).

- **Connecting C_r^+ and C_r^-** : the μ -tangent line segment is internal and crosses the line of centres $\Omega^1 \Omega^2$ (Fig. 10 bottom). Once again, a straightforward geometric analysis shows that the internal μ -tangent line segment exists iff:

$$l(\Omega^1 \Omega^2) \geq 2r \quad (13)$$

and that its length is:

$$l(q_1 q_2) = \sqrt{l(\Omega^1 \Omega^2)^2 - 4r^2 \cos^2 \mu} - 2r \sin \mu \quad (14)$$

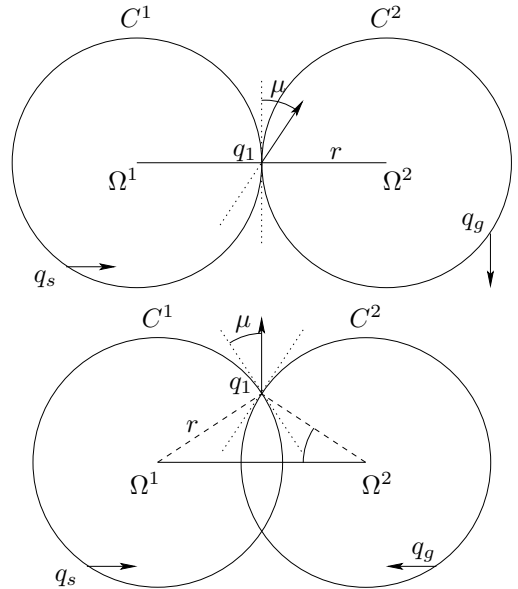


Fig. 11. μ -tangency between two CC Circles: no change of direction of motion (top), change of direction of motion (bottom).

- 2) *CC Circles μ -Tangency*: In Reeds and Shepp's case, two circles associated with two given configurations can also be connected by a circle tangent to both of them (Fig. 9 left). Similarly two CC Circles can be connected thanks

to a third CC Circle. In our case however, the tangency relationship between two CC Circles is different from the classical tangency relationship existing between two circles. Two cases arise depending on whether there is a change of direction of motion when passing from the first CC Circle to the second one.

- **No change of direction of motion:** in this case, the μ -tangency condition between two CC Circles C^1 and C^2 is the classical one. C^1 and C^2 are μ -tangent if they are in the disposition depicted in the top part of Fig. 11. Let q_1 denote the configuration located at the contact point between C^1 and C^2 , and that makes an angle μ with both C^1 and C^2 . Then the path made up of the CC Turn from q_s to q_1 and the CC Turn from q_1 to q_g is a valid CC Path. q_1 belongs to the line of centres $\Omega^1\Omega^2$ and it exists iff:

$$l(\Omega^1\Omega^2) = 2r \quad (15)$$

- **Change of direction of motion:** in this case, the μ -tangency condition between two CC Circles C^1 and C^2 is different. C^1 and C^2 are μ -tangent if they are in the disposition depicted in the bottom part of Fig. 11. Let q_1 denote the configuration located at one of the intersection points between C^1 and C^2 , and that makes an angle μ with both C^1 and C^2 . Then the path made up of the CC Turn from q_s to q_1 (moving forward), and the CC Turn from q_1 to q_g (moving backward) is a valid CC Path. In this case, q_1 does not belong to the line of centres $\Omega^1\Omega^2$ and it can be shown that it exists iff:

$$l(\Omega^1\Omega^2) = 2r \cos \mu \quad (16)$$

3) *Building CC Paths using CC Turns:* there is an infinite number of ways to connect two given configurations with paths combining CC Turns and line segments. It is the same for RS Paths but the key result established by Reeds and Shepp [8] is that the shortest RS Paths belongs to a set of nine families. The conjectured irregularity of the optimal paths for the CC Car would rule out the determination of such families. We settled for heuristically selected families instead. Our first choice was to use the families determined by Reeds and Shepp (we later experimented with an extended set of families but obtained mixed results only [39]). These families are:

(i)(ii)(iii)	CCC or $C CC$ or $CC C$	(17)
(iv)	$CC CC$	
(v)	$C CC C$	
(vi)	$C CSC C$	
(vii)(viii)	$C CSC$ or $CSC C$	
(ix)	CSC	

where C (resp. S) denotes a CC Turn (resp. line segment), and $|$ a change of direction of motion (a cusp point). Note that, by design, the CC Paths above can connect configurations with null curvature only, *ie* for which the front wheels' orientation of the car is null (this restriction is addressed later in §V-B). Given two configurations with null curvature, CC Steer operates by computing the shortest CC Paths among the families (17).

As far as computing the candidate CC Paths between two given configurations is concerned, it should be noted first that all the families (17) are made up of one or more part of type CSC , CC or $C|C$, and that such parts are precisely the type of paths that were built in the two previous sections while studying the various μ -tangency properties (§IV-C.1 and §IV-C.2). It should be noted also that the condition of existence and the characterisation of the CSC , CC and $C|C$ paths stem directly from these μ -tangency properties. Accordingly, the condition of existence of a CC Path of any given type is readily obtained by combining these conditions of existence. Once the existence of a given type of CC Path is ascertained, it can be computed and its arc length can be determined using (12), (14) and the results established in §IV-B.3 on the arc length of a CC Turn.

D. Small-Time Controllability Issues

1) *Topological Admissibility:* as mentioned in §I, the purpose of our steering method CC Steer is to be used within a general motion planning scheme such as the Probabilistic Path Planner [10] or the Ariadne's Clew Algorithm [11] or the Holonomic Path Approximation Algorithm [12]. In order to ensure that the coupling between the planning scheme and the steering method yields a complete (or probabilistically complete) collision-free path planner, the steering method is required to account for the small-time controllability of the system under consideration. In other words, it must be *topologically admissible*, *ie* verify the following topological property [13]:

$$\forall \varepsilon > 0, \exists \eta > 0, \forall (q_1, q_2) \in \mathcal{C}^2, \quad q_2 \in \mathcal{B}(q_1, \eta) \implies \text{Steer}(q_1, q_2) \subset \mathcal{B}(q_1, \varepsilon) \quad (18)$$

where $\mathcal{B}(q, \varepsilon)$ denotes the configuration space ball of size ε centred around q , and $\text{Steer}(q_1, q_2)$ denotes the path from q_1 to q_2 computed by the steering method. In other words, the steering method must be able to connect two η -neighbour configurations with a path that remains in an ε -neighbourhood.

Because of the nature of the CC paths, CC Steer is not topologically admissible. Indeed, no matter how close the start and goal configurations are, the CC Path connecting them includes at least one CC Turn (except if the start and goal configurations are perfectly aligned), and since the length of a CC Turn is lower bounded by $2r \sin \mu$ (*cf* IV-B.3), (18) is violated. The approach we have chosen to make CC Steer topologically admissible is to extend the set of families (17). A new family of CC paths is introduced, henceforth called *topological paths*. They are made up of line segments and clothoid arcs, they do not include CC Turns and are designed so as to verify (18).

2) *Topological Paths:* The topological path between two configurations q_s and q_g is the concatenation of two parts:

- 1) A *reorientation path* between q_s and an intermediate configuration q_j which has the same orientation as q_g and is located on the line passing through q_g with a direction perpendicular to the orientation of q_g (Fig. 12).
- 2) A *lateral path* between q_i and q_g (Fig. 13).

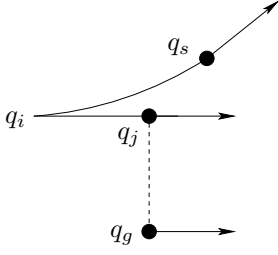


Fig. 12. The reorientation path defined for q_s and q_g : it connects q_s and q_g .

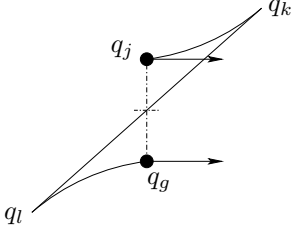


Fig. 13. The lateral path between q_j and q_g .

Reorientation and lateral paths are respectively described in the next two sections whereas appendix C establishes that the topological paths verify the topological property (18).

3) *Reorientation Paths*: The reorientation path defined for two configurations q_s and q_g in the disposition depicted in Fig. 12 has two parts:

- A backward turn starting from q_s made up of two symmetric clothoid arcs to an intermediate configuration q_i which has the same orientation as q_g .
- A forward motion along a line segment until q_j , the configuration located on the line passing through q_g with a direction perpendicular to the orientation of q_g , is reached.

Appendix B details how reorientation paths are computed.

4) *Lateral Paths*: The lateral path between q_j and q_g has three parts (Fig. 13):

- A forward turn made up of two symmetric clothoid arcs connecting q_j and an intermediate configuration q_k .
- A backward motion along a line segment to an intermediate configuration q_l .
- A forward turn made up of two symmetric clothoid arcs connecting q_l to q_g .

There is an infinite number of such paths between q_j and q_g . Two extra constraints are introduced in order to reduce this number to one. First, the lateral path should be symmetric with respect to the midpoint of the line segment connecting q_j and q_g . This constraint forces the line segment part of the lateral path to pass through this midpoint. Second, the shortest lateral path verifying the previous constraint is selected to be the lateral path between q_j and q_g . Appendix A details how lateral paths are computed.

E. CC Steer

In summary, given two configurations with null curvature, CC Steer computes all the existing CC Paths of the different

families (17) plus the topological path connecting them. Then CC Steer selects and returns the shortest candidate.

In order to be complete, CC Steer must be able to compute a connecting path between any pair of arbitrary configurations with null curvature. Given two such configurations, the existence of the CC Path of one of the different families (17) connecting them depends on their disposition (*cf* §IV-C.3). However, by construction, the topological path connecting them always exists (*cf* appendices A and B). Accordingly, CC Steer is complete; it can connect any given pair of configurations with null curvature.

In general, the topological path between two configurations is longer than the shortest CC Paths of the different families (17). However, when the start and goal configurations converge towards one another, the topological path eventually becomes the shortest one and it is selected by CC Steer. CC Steer therefore accounts for the small-time controllability of the CC Car and the coupling between CC Steer and one of the aforementioned general planning scheme yields a complete (or probabilistically complete) collision-free path planner.

F. CC paths and RS Paths

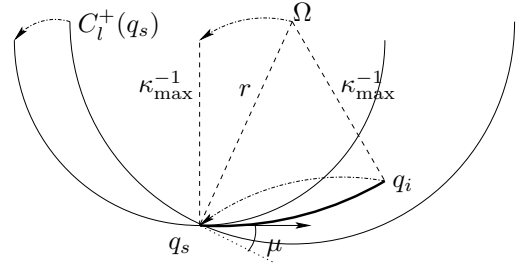


Fig. 14. Convergence of $C_l^+(q_s)$ when σ_{\max} tends to infinity.

Looking back at Fig. 2, it can be noted that the curvature profile of a CC Path would converge towards the curvature profile of a RS Path should σ_{\max} tend to infinity. As a matter of fact, according to Eqs. (6), (7), (8) and (9), it can be verified that, when σ_{\max} tends to infinity:

$$\begin{cases} q_i = (0, 0, 0, \kappa_{\max}) \\ \Omega = (\kappa_{\max}^{-1} \sin \theta_s, \kappa_{\max}^{-1} \cos \theta_s) \\ r = \kappa_{\max}^{-1} \\ \mu = 0 \end{cases}$$

In other words, $C_l^+(q_s)$ becomes a circle of radius κ_{\max}^{-1} tangent to q_s (Fig. 14). This circle is precisely one of the circles that are used to compute RS Paths. Similar convergence affects the other CC Circles $C_r^+(q_s)$, $C_l^-(q_s)$ and $C_r^-(q_s)$. Accordingly, when σ_{\max} tend to infinity, the CC Paths obtained converge towards the corresponding RS Paths.

V. EXPERIMENTAL RESULTS

A. CC paths vs RS Paths

Fig. 15 depicts examples of paths computed by CC Steer. By design, the transition configurations between CC Turns and line segments (marked as cross in Fig. 15) have a null

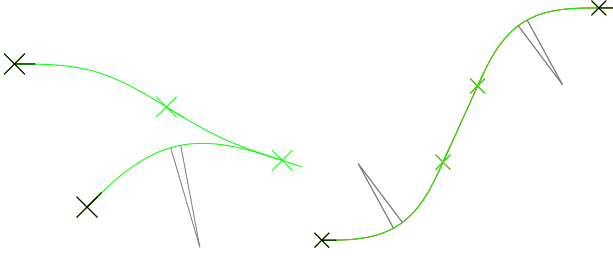


Fig. 15. Examples of CC Paths: $C|CC$ (left) and CSC (right).

curvature (the orientation of the front wheels of \mathcal{A} is null). \mathcal{A} can therefore pass from one part of a CC Path to the next without having to stop in order to reorient its front wheels (this is also true for cusp points).

TABLE I
RS vs CC PATHS' LENGTH (1000 PATHS).

	min.	average	max.	deviation
ratio	1.00253	1.1065	2.45586	0.172188

TABLE II
RS vs CC PATHS' COMPUTATION TIME.

RS (1000 paths)	CC (1000 paths)	average ratio
3.466586 s.	4.483492 s.	1.33

CC Steer has been implemented along with a function computing RS Paths. Comparisons were made regarding the length of the paths and the time required for their computation. The ratio of CC over RS Paths' lengths were computed for a thousand pairs of (start, goal) configurations. The results obtained for $|\sigma_{\max}| = 1$ are summarised in Table I. In most cases (82%), CC paths are only about 10% longer than RS Paths. Similar experiments were carried out for the computation time. The running time of CC Steer is of the same order of magnitude (Table II).

B. Embedding CC Steer in General Planning Schemes

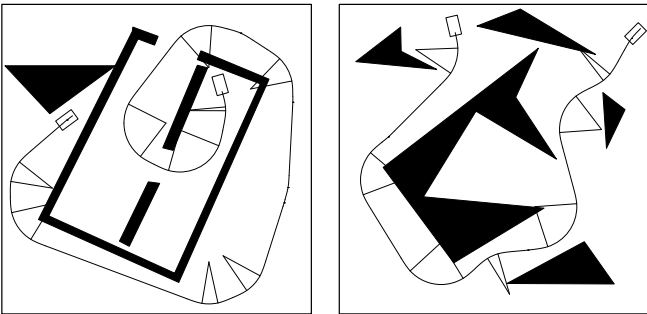


Fig. 16. Examples of collision-free paths (made up of CC Paths) for the CC Car moving forward only [38].

To demonstrate the capabilities of CC Steer for collision-free path planning, we embedded it into two general planning schemes.

First, we used the Probabilistic Path Planner (PPP) [10] to plan collision-free paths for the CC Car moving forward only. In this case, the families of CC Paths considered by CC Steer were restricted to paths without cusp points. Fig. 16 depicts some results obtained for a polygonal environment [38].

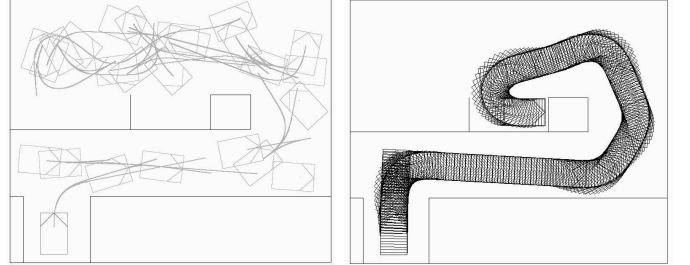


Fig. 17. Collision-free path planning for the CC Car in a parallel parking-like situation [39]

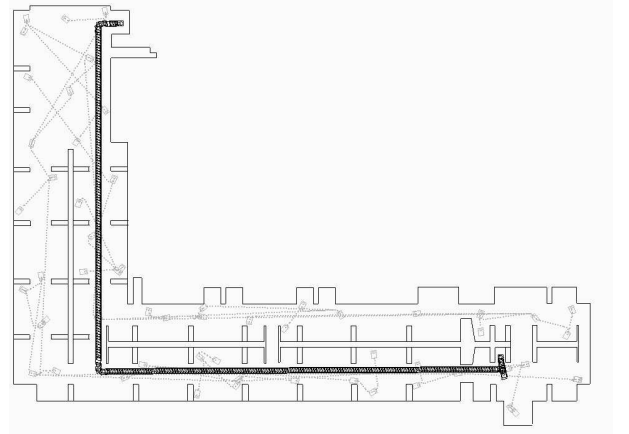


Fig. 18. Collision-free path planning for the CC Car in a parking lot [39]

Second, we used the Ariadne's Clew Algorithm (ACA) [11] to plan collision-free paths for the regular CC Car, *ie* with forward and backward motions. Unlike PPP which is a roadmap-based, ACA is a direct method; it develops a tree rooted at the start configuration until the goal is reached. Fig. 17 depicts the result obtained in a parallel parking-like situation (the exploration tree is depicted on the left part of the figure). Fig. 18 shows a path planning example in the Inria Rhône-Alpes parking lot [39]. The exploration tree is displayed in grey. Note that since CC Steer permits to place tree nodes very far from each other, few tree nodes suffice to cover the whole free space and the resulting path looks quite natural.

In all cases, the collision-free paths resulting from the coupling of CC Steer with a general planning scheme are concatenation of CC Paths. Because CC Paths are designed to connect configurations with null curvature, the transition configurations between two consecutive CC Paths must have null curvature and so do the start and goal configurations. The curvature profile of the overall resulting path is therefore

continuous and a car-like vehicle can move from one CC path to the next without ever having to stop in order to reorient its front wheels (of course, it has to stop at cusp points in order to change its motion direction).

In our opinion, the restriction on the curvature of the start and goal configurations is not so important since an actual car-like vehicle can always reorient its front wheels on the spot (and such a ‘motion’ is collision-free). In other words, to solve a path planning problem between two configurations with non-null curvature, the corresponding path planning problem between the same two configurations with null curvature is solved using a general planning scheme coupled with CC Steer. Then, the collision-free path obtained is completed by two on-the-spot front wheels’ reorientations both at the start and goal configurations.

VI. CONCLUSION

In this paper, we have presented CC Steer, the first steering method for a car-like vehicle that computes paths with continuous curvature, upper-bounded curvature and upper-bounded curvature derivative. CC Steer computes paths made up of line segments, circular arcs and clothoid arcs. It is designed to connect configurations with null curvature, *ie* for which the front wheels’ orientation of the car is null. CC Steer is complete, *ie* it can connect any such pair of configurations, and it verifies a topological property that ensures that the coupling between CC Steer and a general planning scheme yields a complete collision-free path planner. CC Steer does not compute minimal length paths. However, we have shown that it computes, in about the same time, paths whose length is close to the length of the optimal paths for the ‘Reeds and Shepp’ car (as a matter of fact, when the curvature derivative limit tends to infinity, the paths computed by CC Steer tend to the optimal Reeds and Shepp paths).

The coupling of CC Steer with a general motion planning scheme yields a path planner that computes collision-free paths with a continuous curvature profile. A car-like vehicle can therefore follow such a path without ever having to stop in order to reorient its front wheels. Besides, such paths can be followed with a nominal speed which is proportional to the curvature derivative limit. Given that paths with continuous curvature, upper-bounded curvature and upper-bounded curvature derivative can be tracked at high speed with a much greater accuracy by real vehicles (*cf* the experimental results obtained in [35]), the results reported herein fully demonstrate the interest of CC Steer.

APPENDIX

A. Computing Lateral Paths

With reference to Fig. 13, let q_j denote the configuration of null orientation and let L denote the half-line of orientation α anchored at q_j (Fig. 19). Let $q_k(r)$ denote a configuration of orientation 2α located on L at a given distance r . The elementary path (*cf* IV-B.2) connecting q_j and $q_k(r)$ exists and does not violate the upper-bounded curvature constraint (4) iff r is lower bounded (*cf* [36, Property 4] or [37, Theorem 1]):

$$r \geq r_{\min}^1(\alpha) = 2\kappa_{\max}^{-1} \sqrt{2\pi|\alpha|} D(|\alpha|) \quad (19)$$

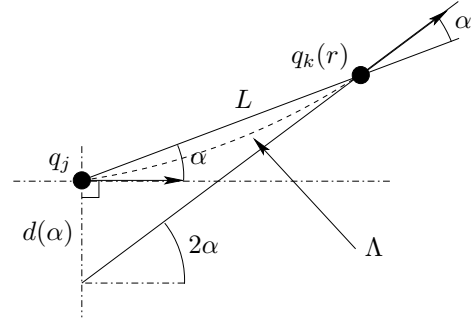


Fig. 19. Computing a lateral path (see Fig. 13).

where D is the real function defined over $[0, \pi]$ as:

$$D(u) = \cos u C_f \left(\sqrt{\frac{2u}{\pi}} \right) + \sin u S_f \left(\sqrt{\frac{2u}{\pi}} \right)$$

with C_f and S_f the Fresnel integrals [40].

The elementary path Λ connecting q_j and $q_k(r)$ is uniquely defined. The sharpness of its clothoid arcs is [36, Property 4] or [37, Eq. (3)]:

$$\sigma(\alpha, r) = 4\pi \operatorname{sgn}(\alpha) \frac{D(|\alpha|)^2}{r^2} \quad (20)$$

and its length is:

$$l(\alpha, r) = 2\sqrt{\frac{2\alpha}{\sigma(\alpha, r)}} \quad (21)$$

Now, the upper-bounded curvature derivative constraint (3) yields an upper bound on the sharpness in (20): $|\sigma(\alpha, r)| \leq \sigma_{\max}$, which yields an additional lower bound on r :

$$r \geq r_{\min}^2(\alpha) = 2\sqrt{\pi\sigma_{\max}^{-1} D(|\alpha|)} \quad (22)$$

Let us define the following function:

$$r(\alpha) = \max\{r_{\min}^1(\alpha), r_{\min}^2(\alpha)\} \quad (23)$$

Given an orientation α , $r(\alpha)$ is the distance of the closest configuration q_k of orientation 2α located on L that can be connected by an elementary path verifying both the upper-bounded curvature constraint (4) and the upper-bounded curvature derivative constraint (3).

The line of orientation 2α passing through q_k intersects the line perpendicular to q_j at a point located at a distance $d(\alpha)$ from q_j :

$$d(\alpha) = \frac{r(\alpha) \sin(2\alpha)}{2 \cos(\alpha) \cos(2\alpha)} \quad (24)$$

Now, to determine the lateral path between two configurations q_j and q_g separated by a distance ε , it suffices to use (24) in order to determine α such that $d(\alpha) = \varepsilon/2$.

B. Computing Reorientation Paths

Let q_g denote a goal configuration of null orientation and let $\delta\theta$ denote the orientation gap between q_g and the start configuration q_s . Let L denote the half-line of orientation $\pi + \delta\theta/2$ anchored at q_s (Fig. 20). As per §A, a backward

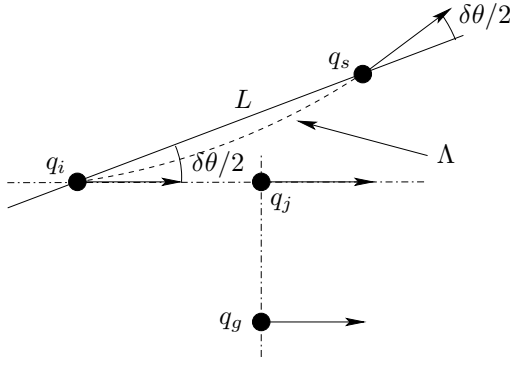


Fig. 20. Computing a reorientation path.

elementary path starting from q_s can connect any configuration of null orientation located on L at a distance greater than $r(\delta\theta/2)$.

Let q_i denote the configuration of null orientation located on L at the minimum distance $r(\delta\theta/2)$. The elementary path Λ connecting q_s and q_i is uniquely defined (cf §A), and so is the configuration q_j which is the intersection between the line of null orientation passing through q_i and the line passing through q_s with a direction perpendicular to the orientation of q_s . The reorientation path defined for q_s and q_g is therefore completely determined.

C. Topological Admissibility of the Topological Paths

Let us assume that q_s and q_g are separated by a distance η (in both position and orientation).

Consider Fig 20, q_i is located at a distance $r(\eta/2)$ from q_s and the length of the elementary path connecting q_s and q_i is $l(\eta/2, r(\eta/2))$. A straightforward geometric analysis shows that the distance between q_i and q_j is $|r(\eta/2) \cos(\eta/2) - \eta|$. It can be concluded then that the reorientation path from q_s to q_j is entirely contained in a ball centered at q_s of radius:

$$r_r = l(\eta/2, r(\eta/2)) + |r(\eta/2) \cos(\eta/2) - \eta|$$

Consider now Figs 13 and 19, it can be shown that the distance between q_j and q_g is $|r(\eta/2) \sin(\eta/2) - \eta|$. Let α be such that $d(\alpha) = |r(\eta/2) \sin(\eta/2) - \eta|/2$, then q_k (resp. q_l) is located at a distance $r(\alpha)$ from q_j (resp. q_g). As for the elementary paths from q_j to q_k , and q_l to q_g , their lengths is $l(\alpha, r(\alpha))$. It can be concluded then that the lateral path from q_j to q_g is entirely contained in a ball centered at q_j of radius:

$$r_l = |r(\eta/2) \sin(\eta/2) - \eta| + l(\alpha, r(\alpha))$$

Accordingly, the topological path between q_s and q_g is entirely contained in a ball centered at q_s of radius:

$$r_t = r_r + r_l$$

When q_s converges towards q_g , ie when η converges towards 0, the orientation α such that $d(\alpha) = |r(\eta/2) \sin(\eta/2) - \eta|/2$ also converges towards 0 (indeed, $\alpha = 0$ is a straightforward solution to $d(\alpha) = 0$). Now, given that both:

$$\lim_{\alpha \rightarrow 0^+} r(\alpha) = 0 \text{ and } \lim_{\alpha, r \rightarrow 0^+} l(\alpha, r) = 0,$$

we have r_t that converges towards 0. Accordingly, thanks to the topological paths, CC Steer can access a neighbourhood of a configuration q_s without escaping a ball centered at q_s (no matter how small the ball). It accounts for the small-time controllability of \mathcal{A} .

REFERENCES

- [1] J.-P. Laumond, "Feasible trajectories for mobile robots with kinematic and environment constraints," in *Proc. of the Int. Conf. on Intelligent Autonomous Systems*, Amsterdam (NL), Dec. 1986, pp. 346–354.
- [2] J.-P. Laumond, Ed., *Robot motion planning and control*, ser. Lecture Notes in Control and Information Science. Springer, 1998, vol. 229.
- [3] J. Barraquand and J.-C. Latombe, "On non-holonomic mobile robots and optimal maneuvering," *Revue d'Intelligence Artificielle*, vol. 3, no. 2, pp. 77–103, 1989.
- [4] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ (US), May 1989, pp. 2–7.
- [5] T. Fraichard, "Smooth trajectory planning for a car in a structured world," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, Sacramento, CA (US), Apr. 1991, pp. 318–323.
- [6] P. Svestka and M. H. Overmars, "Probabilistic path planning," Dept. of Computer Science, Utrecht University (NL), Technical Report UU-CS-1995-22, May 1995.
- [7] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–517, 1957.
- [8] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [9] J.-D. Boissonnat, A. C  r  zo, and J. Leblond, "A note on shortest paths in the plane subject to a constraint on the derivative of the curvature," Inst. Nat. de Recherche en Informatique et en Automatique, Rocquencourt (FR), Research Report 2160, Jan. 1994.
- [10] L. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robotics and Automation*, vol. 12, pp. 566–580, 1996.
- [11] E. Mazer, J.-M. Ahuactzin, and P. Bessiere, "The Ariadne's Clew Algorithm," *Journ. of Artificial Intelligence Research*, vol. 9, pp. 295–316, July-December 1998.
- [12] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for non-holonomic mobile robots," *IEEE Trans. Robotics and Automation*, vol. 10, no. 5, pp. 577–593, Oct. 1994.
- [13] S. Sekhavat and J.-P. Laumond, "Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems," *IEEE Trans. Robotics and Automation*, vol. 14, no. 5, pp. 671–680, Oct. 1998.
- [14] H. J. Sussmann, "The Markov-Dubins problem with angular acceleration control," in *IEEE Conf. on Decision and Control*, San Diego, CA (US), Dec. 1997.
- [15] K. Komoriya and K. Tanie, "Trajectory design and control of a wheel-type mobile robot using B-spline curve," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Tsukuba (JP), Sept. 1989, pp. 398–405.
- [16] A. Takahashi, T. Hongo, and Y. Ninomiya, "Local path planning and control for AGV in positioning," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Tsukuba (JP), Sept. 1989, pp. 392–397.
- [17] W. L. Nelson, "Continuous curvature paths for autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, Scottsdale, AZ (US), May 1989, pp. 1260–1264.
- [18] R. Liscano and D. Green, "Design and implementation of a trajectory generator for an indoor mobile robot," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Tsukuba (JP), Sept. 1989, pp. 380–385.
- [19] Y. Kanayama and B. I. Hartman, "Smooth local path planning for autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, Scottsdale, AZ (US), May 1989, pp. 1265–1270.
- [20] A. Piazzi and C. Guarino Lo Bianco, "Quintic G^2 -splines for trajectory planning of autonomous vehicles," in *Proc. of the IEEE Intelligent Vehicles Symp.*, Dearborn, MI (US), Oct. 2000, pp. 198–203.
- [21] H. Delingette, M. H  bert, and K. Ikeuchi, "Trajectory generation with curvature constraint based on energy minimization," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, Osaka (JP), Nov. 1991, pp. 206–211.

- [22] A. E. Bryson and Y. C. Ho, *Applied optimal control*. Hemisphere Publishing, 1975.
- [23] C. Fernandez, L. Gurvits, and Z. L. Li, "A variational approach to optimal nonholonomic motion planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, Nagoya (JP), May 1995, pp. 680–685.
- [24] S. Sekhavat, "Planification de mouvements sans collisions pour des systèmes non holonomes," Inst. Nat. Polytechnique de Toulouse (FR)," Thèse de doctorat, 1995.
- [25] G. Lafferrière and H. Sussmann, "A differential geometric approach to motion planning," in *Nonholonomic Motion Planning*, Z. Li and J. Canny, Eds. Kluwer, 1993, pp. 235–270.
- [26] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: steering using sinusoids," *IEEE Trans. on Automatic Control*, vol. 38, pp. 700–716, 1993.
- [27] P. Rouchon, M. Fliess, M. Lévine, and P. Martin, "Flatness, motion planning and trailer system," in *IEEE Proc. of the IEEE Conf. on Decision and Control*, San Antonio, TX (US), Dec. 1993, pp. 2700–2705.
- [28] J.-P. Laumond, S. Sekhavat, and F. Lamiriaux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot motion planning and control*, ser. Lecture Notes in Control and Information Science, J.-P. Laumond, Ed. Springer, 1998, vol. 229, pp. 1–53.
- [29] A. Bellaïche, J.-P. Laumond, and M. Chyba, "Canonical nilpotent approximation of control systems: application to nonholonomic motion planning," in *Proc. of the IEEE Int. Conf. on Decision and Control*, San Antonio, TX (US), 1993.
- [30] D. Tilbury, R. Murray, and S. Sastry, "Trajectory generation for the n -trailer problem using Goursat normal form," *IEEE Trans. on Automatic Control*, vol. 40, no. 5, pp. 802–819, 1995.
- [31] S. Monaco and D. Normand-Cyrot, "An introduction to motion planning under multi-rate digital control," in *Proc. of the IEEE Int. Conf. on Decision and Control*, Tucson, AZ (US), 1992, pp. 1780–1785.
- [32] S. Sekhavat, F. Lamiriaux, J.-P. Laumond, G. Bauzil, and A. Ferrand, "Motion planning and control for Hilare pulling a trailer: experimental issues," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM (US), Apr. 1997.
- [33] F. Lamiriaux and J.-P. Laumond, "Smooth motion planning for car-like vehicles," *IEEE Trans. Robotics and Automation*, vol. 17, no. 4, pp. 498–502, Aug. 2001.
- [34] E. Degtiariova-Kostova and V. Kostov, "Irregularity of optimal trajectories in a control problem for a car-like robot," Inst. Nat. de Recherche en Informatique et en Automatique, Research Report 3411, Apr. 1998.
- [35] A. Scheuer and C. Laugier, "Planning sub-optimal and continuous-curvature paths for car-like robots," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, Victoria, BC (CA), Oct. 1998, pp. 25–31.
- [36] A. Scheuer, "Planification de chemins à courbure continue pour robot mobile non-holonyme," Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble (FR), Jan. 1998.
- [37] A. Scheuer and T. Fraichard, "Collision-free and continuous-curvature path planning for car-like robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, Albuquerque, NM (US), Apr. 1997, pp. 867–873.
- [38] —, "Continuous-curvature path planning for car-like vehicles," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, Grenoble (FR), Sept. 1997, pp. 997–1003.
- [39] T. Fraichard and J.-M. Ahuactzin, "Smooth path planning for cars," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Seoul (KR), May 2001.
- [40] M. Abramovitz and I. A. Stegun, Eds., *Handbook of mathematical functions*. Dover Publishing Inc., 1965.